contribution-guide

Contributing Guide Thanks for your interest in contributing to Play's documentation! ■ Check the guidelines below for suggestions and requirements before submitting your contribution. Development Workflow ■ Previewing Changes ■ In order to run the documentation locally, install the dependencies and run the development server: npm install npm start Linting Documentation ■ This repository uses Prettier , an opinionated code formatter, in order to keep consistent formatting throughout the documentation. Run the following command to automatically fix all formatting, and then push any changes: npm run lint Spell Check ■ This repository uses cspell , a spell checker for code, to automatically flag any spelling errors. Run the following command to see any spelling errors: npm run spellcheck [!NOTE]

Any spelling errors will need to be fixed manually. There are various ways to ignore words or sections that were flagged erroneously. These are listed below. Ignoring words ■ To ignore: A specific word , add it to the following file: cspell-wordlist.txt A directory or anything matching a regular expression , update the following file: cspell.json For example, we don't want to flag anything inside of code ticks (`) or code blocks (```), so there are regular expressions added to ignore anything inside of these. An entire line , add the following comment above it: <!-- cspell:disable-next-line --> Multiple lines , add comments above and below the lines to be ignored: <!-- cspell:disable --> < p > Everything inside of these comments will be ignored by the spell checkr. Proofread your own words carefully. </ p > <!-- cspell:enable --> [!IMPORTANT]

You need to have line breaks between the cspell comments and any HTML elements, otherwise the build will error with Module build failed . Tips ■ Before adding a word or section to be ignored, see if there is a way to make it pass the spell check. Technical terms that are part of an API may need to be wrapped in code formatting. For example, the word keydown is flagged as an unknown word by the spell checker, but this is a Web API event . We can wrap any mentions of keydown in two backticks (`keydown`) in order to avoid it being flagged by the spell checker. Comments disabling the next line or entire sections of documentation are useful for making the spell checker ignore people's names. In general, we should try to avoid ignoring words unless they are technical terms that are used throughout the documentation and wouldn't necessarily make sense formatted as code. Using VS Code on Windows ■ The Play docs were originally built in a Mac-based environment, so Mac-focused linting rules apply when committing changes. To contribute on Windows, do the following: Configure VS Code to read/save files using line breaks (LF) instead of carriage returns (CRLF). Set it globally by navigating to: Settings -> Text Editor -> Files -> Eol. Set to \n . Check that the Git setting core.autocrlf is set to false : run git config -l | grep autocrlf . Switch it to false using: git config --global core.autocrlf false . Project Structure ■ Play's documentation is built using Docusaurus . The content is either written or generated as Markdown. Directories ■ scripts/ - build scripts used to generate markdown or json files src/ - source code and content of the docs components/ - components used across the site global/ - components used globally page/ - components used on a single page or in a limited scope styles/ - global styles and variables components/ - styles split out into the components they target static/ demos/ - self-contained demos, optionally presented by pages via demoUrl YAML frontmatter versioned_docs/ - versions of the docs created by the docusaurus versioning command versioned_sidebars/ - versions of the docs sidebars created by the docusaurus versioning command Authoring Content ■ The content of the Play docs is written as Markdown in docs/ . Each Markdown file corresponds to a route unless explicitly changed in the frontmatter. /docs/ => src/pages/index.md /docs/intro/cli => src/pages/intro/cli.md /docs/theming/advanced => src/pages/theming/advanced.md /docs/theming => src/pages/theming.md You can make copy edits to the site by editing the Markdown files directly on GitHub . In your pull request, please explain what was missing from or inaccurate about the content. Reporting Issues ■ Before submitting an issue to the Play docs repo, please search existing issues to avoid duplicate reports. If the issue you're reporting is a bug, please be sure it is an issue with the Play docs themselves and not the subject of the documentation. With your report, please provide: Steps to reproduce Expected behavior OS and browser versions If possible, a demo repo or

CodePen/CodeSandbox Pull Request Guidelines ■ When submitting pull requests, please keep the scope of your change contained to a single feature or bug. When in doubt, err on the side of smaller pull requests. If your pull request is a new feature, we would recommend opening an issue first to come to an agreement about the feature before putting in significant time. Deploying ■ The Play documentation's master branch is deployed automatically and separately from the Play site itself. The Play site then uses a proxy for paths under /docs to request the deployed documentation.